AF IPW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Allavarpu, et al. §

Serial No. 09/556,068 §

Filed: April 21, 2000 §

For: Secure Access to Managed Network Objects using a Configurable Platform-Independent Gateway §

Group Art Unit: 2154 §

Examiner: Patel, Haresh N. §

Atty. Dkt. No.: 5181-48400 P4500 §

CERTIFICATE OF MAILING
37 C.F.R. § 1.8

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below:

Robert C. Kowert
Name of Registered Representative

July 12, 2005
Date

Signature

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir/Madam:

Further to the Notice of Appeal filed May 12, 2005, Appellants present this Appeal Brief. Appellants respectfully request that the Board of Patent Appeals and Interferences consider this appeal.

I. REAL PARTY IN INTEREST

As evidenced by the assignment recorded at Reel/Frame 010940/0123, the subject application is owned by Sun Microsystems, Inc., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and now having its principal place of business at 4150 Network Circle, Santa Clara, CA 95054.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-63 stand finally rejected. The rejection of claims 1-63 is being appealed. A copy of claims 1-63 as currently pending is included in the Claims Appendix herein below.

IV. STATUS OF AMENDMENTS

An amendment to claims 61-63 was filed on April 8, 2005 after the final rejection. In the Advisory Action dated April 25, 2005, the Examiner refused to enter this amendment, even though the amendment reflected the language suggested by the Examiner on pages 19 and 20 of the Final Office Action. Additionally, two further amendments were submitted by facsimile on July 8 and July 11, 2005, respectively. The July 8 amendment included amendments to claims 39-57, 60 and 63 reciting a tangible, computer accessible medium rather than a carrier medium. The July 11 amendment repeats the amendments from July 8 to claims 39-57, 60 and 63, but also includes additional amendments, inadvertently omitted from the amendment submitted on July 8, to claims 61-63 correcting minor antecedent basis issues, as discussed in the Arguments section below. The Examiner has not yet indicated whether or not these amendments will

be entered. As such, the Claim Appendix included herein below reflects the state of the claims prior to submission of the amendments mentioned above.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Many types of devices may be managed over a network, such as printers, scanners, phone systems, copiers, and many other devices and appliances configured for network operation. Typically, such devices are managed via requests and events. A request is a message sent to a managed object. A request may be sent by a manager application to a managed object to query the object about a particular parameter associated with the object. A request may also be sent to a managed object to modify a parameter of the object. An event is a message that may originate with a managed object. Events may be sent by managed objects to signal some change of state of the managed object, or to communicate information about the managed object.

Independent claim 1 is directed toward a network management system including a gateway coupled to a plurality of managed objects and which is configured to deliver events generated by the managed objects to managers or to deliver requests generated by the managers to the managed objects. For example, a gateway may reside between applications and an enterprise manager, which may provide an interface through which requests and events may be communicated with managed objects on the network. In one embodiment, a gateway may be coupled to an Object Request Broker (ORB) via communications method such as the Internet Inter-Object Protocol (IIOP) and to an enterprise manager via a proprietary or platform-dependent interface such as the Portable Management Interface (PMI) from Sun Microsystems, Inc. A gateway may include any of various components such as a Metadata Gateway, an Event Gateway or a Request Gateway. Events may be dispatched, such as to clients, managers, or applications, using the gateway. (*see, e.g.* FIGs. 2, 3, 4, 6, 7, 8; page 11, lines 4 – 20; page 20, line 27 – page 21, line 3).

The network management system of claim 1 also includes a platform-independent

interface to the gateway, wherein the gateway is configurable to communicate with the managers through the platform-independent interface to deliver the events or requests. Events may include notifications, warnings, or alarms concerning a status or occurrence relating to the managed object or a corresponding device. For example, in one embodiment, an event may be generated if usage of a particular resource reaches a particular threshold. Manager applications may send requests and receive responses or events as part of a platform-independent interface to a gateway, such as through the use of Interface Definition Language (IDL), Internet Inter-Object Protocol (IIOP), CORBA, or through the use of proprietary protocols, according to various embodiments. A gateway may also translate requests from one format or protocol to another. In one embodiment, events may be dispatched to clients, such as managers or manager applications, using an event gateway. In some embodiments, an event gateway may be capable of filtering events according to various criteria or through event subscriptions. Thus, a gateway may determine which clients are to receive what events and route the events to the appropriate clients. (*see, e.g.* FIGs. 2, 3, 7, 8, 10, 15; page 21, line 26 – page 22, line 13; page 25, lines 12 – 21; page 45, lines 12- 30).

The gateway of a network management system may be configurable to provide object-level access control between the managers and the managed objects to receive the events from or to send the requests to the managed objects. The object-level access control may be provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. In other words, manager access to managed objects may be granted at the individual object level. For instance, a manager may initially be authenticated to the gateway, such as by username and password or other validation information, and may be further authenticated for each event type at the individual object level. For example, the gateway may check the manager's privilege to receive events from a given managed object and/or for each managed object. (*see, e.g.* FIGs. 4, 8, 12, 13, 14 and 15; page 12, line 24 – page 13, line 4; page 14, lines 2-12; page 26, line 15 – page 27, line 7; page 32, line 23 – page 33, line 16).

Independent claim 20 is directed toward a network management method including sending an identity of a user of a manager application to a gateway. The gateway may be configurable to communicate with the manager application through a platform-independent interface. A client, such as a manager application, may provide a user's identity as part of authentication criteria, which may be encrypted for security. As described above, a manager may be authenticated to the gateway, such as by username and password or via other validation information. The authentication or validation information may be represented in a user profile accessed using the user's identity provided to the gateway. The user identity may also provide access to event subscription information when determining whether a manager application has access to a particular object and/or event. Additionally, in some embodiments, user information may be included in each request sent through the gateway. For more information regarding communicating through a platform-independent interface, please see the above discussion regarding claim 1. (*see, e.g.* page 12, lines 7-22; page 13, lines 20-29).

Additionally, the network management method of claim 20 also includes determining, on a managed object level, whether or not a manager application is allowed to receive an event generated by one of the managed objects or to send a request to the one of the managed objects as a function of the identity of the user of the manager application. Access for the manager application to receive the event or send the request is approved or denied for the managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects. As discussed above, a manager application may be authenticated for each event type at the individual object level and the gateway may check the manager's privilege to receive events from a given managed object and/or for each managed object. Furthermore, the network management method includes delivering the event to the manager application or the request to the managed object if the manager access is approved. (FIGs. 12, 13, 14 and 15; page 12, line 24 – page 13, line 4; page 14, lines 2-12; page 26, line 15 – page 27, line 7; page 32, line 23 – page 33, line 16).

Independent claim 39 is directed to a medium including program instructions for network management that are computer-executable to perform the network management method described above regarding claim 20. Please refer to the discussion of claim 20 above for more details regarding the network management method.

Independent claim 58 is directed to a network management system similar to the network management system described above regarding claim 1 except that the system recited in claim 58 includes a manager using a request Service Access Point (SAP) for requests and responses. A request Service Access Point, or RequestSAP, as opposed to a regular Service Access Point, may be configured to allow the insertion of a user name in the request message, such as to enforce object-level access control. (*see, e.g.* page 14, lines 2 – 12; page 33, lines 2 – 16; page 36, lines 1-7).

Independent claim 59 is directed to a network management method similar to the method described above regarding claim 20 except that the method recited in claim 59 includes the manager application using a request Service Access Point (SAP) for requests and responses. Please refer to the discussions above regarding claims 20 and 58 for more information on the use of a request Service Access Point (RequestSAP) for sending and receiving requests and responses.

Independent claim 60 is directed toward a medium including program instructions for network management that are computer-executable to perform the method recited by independent claim 59. Please refer to the above discussion of claim 59 for more details.

Independent claim 61 is directed toward a network management system similar to the network management system described above regarding claim 1, but in which the gateway uses a singleton SAP object that shares all ProxyAgents through which a manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control. As noted above, a request Service Access Point, or RequestSAP, may be a singleton object that may be shared by all ProxyAgents. A gateway may create a Request SAP to send a request and the

RequestSAP may be configured to allow the insertion of a user name in the request message, such as to enforce object-level access control, as described above. Additionally, a ProxyAgent may provide a platform-independent interface, such as via exposed IDL methods, to allow registration or subscription of events by object class, name, event type, etc. (*see, e.g.* page 14, lines 2 – 12; page 23, line 28 – page 24, line 11; page 33, lines 2 – 16; page 36, lines 1-7).

Independent claim 62 is directed toward a network management method similar to the network management method described above regarding claim 20, but including wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which the manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control. Please see the discussions above regarding claims 20 and 61 for more details regarding a gateway using a singleton SAP object that shares ProxyAgents.

Independent claim 63 is directed toward a medium including program instructions for network management that are computer-executable to perform the method described above regarding claim 62. Please refer to the discussion of claim 62 above for more details.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 58-63 stand finally rejected under 35 U.S.C. § 112, second paragraph, as indefinite.

2. Claims 1-57 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Barker et al. (U.S. Patent 6,363,421) (hereinafter "Barker").

3. Claims 58-63 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Vuong et al. (U.S. Patent 6,430,578) (hereinafter "Vuong").

4. Claims 58-63 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Spencer (U.S. Patent 6,253,243) (hereinafter "Spencer").

5. Claims 58-63 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Barker.

VII. ARGUMENT

First Ground of Rejection:

Claims 58-63 stand finally rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 58:

The Examiner asserts that claim 58 includes the phrase "the manager access", but does not provide sufficient antecedent basis. However, claim 58 does not include the phrase "the manager access" as the Examiner contends. Thus, the § 112, second paragraph, rejection of claim 58 is clearly improper.

Claim 59:

The Examiner rejected claim 59, arguing that there is insufficient antecedent basis for the phrase, "the manager access." However, claim 59 includes the phrases "access for the manager application" and "the manager application is granted access", thus making the subsequent use of the phrase, "the manager access" clear and fully understandable to one of ordinary skill in the art. As noted in § 2173.05(e) of the M.P.E.P, "[i]f the scope of a claim would be reasonable ascertainable by those skilled in the art, then the claim is not indefinite." As one skilled in the art would easily ascertain that the phrase, "the manager access" refers to the access granted the manager application, as recited earlier in the claim, claim 59 is clearly not indefinite.

Claim 60:

As with claim 59, discussed above, the Examiner rejected claim 60, arguing that there is insufficient antecedent basis for the phrase, "the manager access." However, as with claim 59, claim 60 includes the phrases "access for the manager application" and "the manager application is granted access", thus making the subsequent use of the phrase, "the manager access" clear and fully understandable to one of skill in the art. Appellants submit that one skilled in the art would easily ascertain that the phrase, "the manager access" refers to the access granted the manager application, as recited earlier in the claim. As such, claim 60 is not indefinite.

Claims 61-63:

The Examiner rejected claims 61-63, asserting that there is insufficient antecedent basis for the phrases, "the insertion of the user name" and "the request message to enforce object-level access control." An amendment to claims 61-63 was previously submitted on April 8, 2005, along with other amendments, to address this antecedent basis issue. However, the Examiner refused to enter the amendments. Additionally, an amendment was filed via facsimile on July 11, 2005 including amendments to claims 61 – 63, correcting the Examiner's noted antecedent issues.

Even without amendment, Appellants submit that claims 61-63 are clear and fully understandable in their current state. Not only are the respective scopes of these claims easily ascertainable to one of ordinary skill in the art, but the claims recite no other features or limitations that could be confused or made indefinite because of the minor antecedent issues pointed out by the Examiner. Use of the article 'the' before 'insertion', "user name" and 'request message' does not render anything in the claims indefinite since the claims include no other instances of 'insertion', "user name" or 'request message'. Appellants submit that claims 61-63 define their respective subject matter with a reasonable degree of particularity and distinctness. As stated in § 2173.02 of the M.P.E.P, some latitude in the manner of expression and in the aptness of terms should be

permitted even though the claim language is not as precise as the Examiner might desire.

Furthermore, the language currently rejected as indefinite by the Examiner, namely, "the insertion of the user name" and "the request message" exactly reflects the language that the Examiner himself proposed in a facsimile dated May 25, 2004. Given that these terms must have been clear and understandable to the Examiner when the Examiner himself proposed this language, Appellants fail to see how the Examiner can now assert that this language is indefinite.

Second Ground of Rejection:

Claims 1-57 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Barker et al. (U.S. Patent 6,363,421) (hereinafter "Barker"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 4-8, 16 and 17:

Regarding claim 1, Appellants submit that Barker does not anticipate a gateway configurable to provide *object-level access control* between the managers and the managed objects, wherein the object-level access control is provided *at the individual object level* so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. Instead, Barker discloses a system for "access control based on client name and password" (Barker, column 8, lines 45-46). Barker describes this as "a method of *client based* access control of network elements" (emphasis added, Barker, column 30, lines 45-46). Further, Barker summarizes his access control features as "the *client based access control* ... provides a means to restrict access on a *command/client basis*", and does not describe his access control features as restricting access at the object level (emphasis added, Barker, column 31, lines 10-12).

The Examiner cites a passage from Barker (column 23, line 55 – column 26, line 10) describing a set of procedures whereby a client may register to receive notification when managed object attribute values change. The Examiner specifically quotes one line that states, "[n]ote that if more than one attribute has changed for a managed object instance, the changes will be grouped and delivered to each registered client on a managed object instance basis" (Barker, column 26, lines 6-10). Although this portion of Barker teaches clients receiving attribute updates from individual managed objects, and hence object-level notification, it does not teach object-level *access control.* Barker explicitly teaches providing client-based access control at the start of a client session (see Barker, column 30, lines 47-52), while not requiring further authentication or access control based upon which managed objects the client wishes to access. The portion of Barker cited by the Examiner has absolutely nothing to do with *access control* provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

In the Response to Arguments section of the Final Action, the Examiner contends that Barker's use of a managed object identifier for network elements teaches object-level access control and cites Figure 6 of Barker. However, Figure 6 of Barker only teaches that a combination of a managed object class code and an instance identifier defines the managed object identifier. Figure 6 illustrates nothing regarding object level access control. Instead, Figure 6 is merely a listing of terms. Barker specifically states, "FIG. 6 is a table of terms associated with the managed object model" (Barker, column 3, lines 12-13). The mere use of the term "managed object identifier" cannot properly be construed as to disclose, teach, or even suggest controlling access to managed objects via object-level access control as recited in claim 1. Furthermore, the Examiner seems to be implying that if Barker's system includes any mechanism to address individual managed objects, Barker's system necessarily includes object-level access control. This is plainly incorrect. Object-level *addressing* and object-level *access control* are two very different things and object-level addressing does not imply object-level access control.

As the Examiner notes, "[e]ach managed object class requires the session identifier as a parameter to each public method" (Barker, column 30, lines 56-58). The session identifier included as a parameter in each public method allows a managed object class to validate the current session – i.e. to ensure that the client has registered with the server and that the session is currently valid. Barker does not teach a client presenting a user name, password or other authentication credentials when registering for object attribute update notification. In other words, Barker does not teach any access control at the object level. Instead, Barker teaches client-level access control where the client must only provide the session ID, object instance identifier, a set of desired attribute codes, and a callback function when registering for attribute update notifications (see Barker, column 25, lines 23-30). In response, the Examiner, in the Response to Arguments section of the Final Action, asserts that, "a client presenting a user name, password or other authentication credentials when registering for object attribute update notification" is not recited in Appellants' claims. However, the Examiner has misunderstood Appellants' argument. Appellant is arguing that Barker only requires client authentication credentials when initially registering for a session and that Barker does not teach or require a client to present credentials, such as user name, password, etc, when registering for object attribute update notifications, as would be required if Barker actually taught object-level access control. The fact that Barker fails to include any mechanism that could be used to provide access controls on an object-level basis clearly shows that Barker fails to disclose any form of object-level access control. Thus, Barker teaches that a managed object class relies upon the server to perform client authentication by requiring a client to only include a valid session identifier in public method calls without providing or requiring any authentication at the managed object level.

Additionally, Barker teaches that a client can specify a range of managed object instance identifiers, or even *request all instances* in a managed object call through the managed object instance identifier parameter (Barker, column 25, lines 27-28). Hence, Barker teaches that once a client has been properly authenticated at the start of a session, that client may then register for attribute update notification for a number of managed objects through a single call. Such functionality is clearly not compatible with object-

level access control, and thus Barker clearly teaches away from object-level access control, wherein the object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

In response to the Appellants' previous arguments, the Examiner argues that Barker uses "a naming service that provides individual object level access control so that an agent is granted access to an object on the network to support the IIOP protocol" citing column 8, line 53 – column 9, line 19 and column 7, lines 47-63. Appellants note, however, that these passages of Barker only refer to his use of EMAPI, CORBA, Java, C++, and SNMP, but fail to mention anything regarding any sort of access control for any portion of Barker's system. The Examiner has not cited any particular portion in Barker that describes the features the Examiner is attributing to Barker's system. In fact, the Examiner is incorrectly assuming that Barker's use of CORBA and the IIOP protocol includes object level access control such that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

The Examiner also cites Barker as teaching, "access permissions associated with the session are examined before authorizing client execution (e.g. remove operation)" (parenthesis in original) (Barker, column 30, lines 58-60). However, this portion of Barker is clearly referring to ensuring that the client has started a valid session with the server. In fact, Barker, referring to the same remove operation, clearly states, "[a]s with any other client requests, the *client must have created a session prior to performing this operation.*" (Emphasis added, Barker, column 22, lines 51-53). Thus, Barker is clearly referring to client-level access control, not object-level access control. The portion of Barker cited by the Examiner actually supports Appellants' argument.

Barker clearly does not teach object-level access control between the managers and the managed objects. The Examiner contends, in the Response to Arguments section of the Final Action, that Barker's use of a managed object identifier for network elements

teaches object-level access control and again cites Figure 6 of Barker. However, Figure 6 of Barker only teaches that a combination of a managed object class code and an instance identifier defines the managed object identifier. As noted above, Figure 6 does not mention anything regarding object level access control. The mere use of an managed object identifier does not disclose, teach, or even suggest controlling access to managed objects via object-level access control. Object identifiers are used in all types of systems regardless of what type of access control is provided. The Examiner's arguments are completely irrelevant to object-level access control.

Claim 2:

Regarding claim 2, Barker fails to teach that the gateway is configurable to determine whether each of the managers is authorized to communicate with each of the managed objects. Instead, Barker teaches the use of a single service object "to provide services for *a class* of managed objects" (emphasis added, Barker, column 14, lines 42-43) and that the EM server "will implement one application-specific service object for each type of physical or logical resource to be managed" (underlining added, Barker, column 39, lines 60-62). Appellants assert that access control on a command/client basis while using a single service object for *each class* of managed object actually teaches away from determining on a managed object level whether or not the manager application is allowed to send a request to the managed object.

In response, the Examiner cites various pieces of Barker's system in the Response to Arguments section of the Final Action. Specifically, the Examiner refers to Barker's element management server in Figure 1A, software modules of Figures 3 and 4, the term 'user session' of figure 6 and 'network elements' of figure 1C. However, the Examiner completely failed to cite any portion of Barker that actually teaches or suggests that the various, disparate, elements of Barker's system cited by the Examiner actually perform or teach a gateway configurable to determine whether each of the managers is authorized to communicate with each of the managed objects. The fact that Barker includes various system elements that may correspond to certain elements recited in claim 2 does not

imply that they are arranged as in claim 2 nor that they perform specific limitations as recited in claim 2.

Additionally, Barker discloses client-based access control that provides a means to restrict access on a command/client basis (Barker, column 31, lines 10-12). Hence, Barker teaches access control based on a command/client basis, not a managed object basis and thus fails to disclose a gateway that is configurable to determine whether each of the managers is authorized to communicate with each of the managed objects.

Claim 3:

Regarding claim 3, contrary to the Examiner's assertion, Barker fails to teach a gateway configurable to authenticate the managers to receive events from or to send requests to the managed objects as a function of the identity of the managed object. As the Examiner states, Barker teaches the use of basic server authentication, SSL, and web server administration including client name and password for access control (Barker, column 8, lines 31-54). Further, Barker discloses client based access control that provides a means to restrict access on a command/client basis (Barker, column 31, lines 10-12). However, basic server authentication and SSL using client names and passwords do not imply authenticating managers *as a function of the identity of the managed object.*

In response, the Examiner cites various pieces of Barker's system in the Response to Arguments section of the Final Action. Specifically, the Examiner refers to Barker's element management server in Figure 1A, software modules of Figures 3 and 4, and the term "notification" in figure 6. However, the Examiner completely failed to cite any portion of Barker that actually teaches or suggests that the various elements cited by the Examiner actually include or teach a gateway configurable to authenticate the managers to receive the events from or to send the request to the managed objects as a function of the identity of the managed object. The fact that Barker includes various system elements that may correspond to certain elements recited in claim 3 does not imply that

they are arranged as in claim 3 nor that they perform specific limitations as recited in claim 3.

Claim 8:

In regard to claim 8, Barker fails to teach that the managed objects comprise one or more objects *corresponding* to a telephone network. In contrast, Barker discloses a system client that is connected to a network element and element management system client *through* a public switched telephone network (Barker, column 3, lines 48-53). Additionally, Barker teaches the use of a telephone system network through the Internet and a telephonic link for a system client to connect to the system server (Barker, column 3, lines 54-62). The Examiner argues that Barker's use of the phrase "network elements of a telecommunication network" (See, Barker, Title, and brief descriptions of FIG 1A, 1B, and 1C, column 2, lines 50-65) imply that one or more of Barker's network elements correspond to a telephone network. Appellants submit, however, that Barker is referring to network element residing *on* a telecommunications network, not an object *corresponding to* a telephone network. For instance, when discussing FIG. 1B, Barker describes his system as a "method for managing the network element 14 *in* a telephonic network" and continues, "[n]etwork element 14 is connected *through* a telephonic computer network 35 to a computer internet 36" (emphasis added, Barker, column 3, lines 53-58).

In response, the Examiner, in the Response to Arguments section, cites network element 14 of Figure 1A. However, as noted above, network element 14 is not illustrated as corresponding to a telephone network, but rather network element 14 is illustrated as coupled to and communicating over a telephone network. Hence, Barker discloses *using* a telephonic connection between clients and servers but fails to disclose anything regarding managed objects comprising objects *corresponding to* a telephone network. This is made clear when figures 1A, 1B and 1C are viewed together. It is very clear that Barker is illustrating that fact that his system may be implemented (e.g. his element management server may communicate with network elements) over various types of

communication networks, such as public switched telephone network 33 (Figure 1A), telephonic system network 37 (Figure 1B), and a local area network (Figure 1C).

None of the managed objects in Barker *correspond to a telephone network* themselves, but instead communicate using a telephone network. Thus, Barker does not teach wherein the managed objects comprise one or more objects corresponding to a telephone network.

Claim 10:

Regarding claim 10, the Examiner contends that Barker teaches a gateway that is configurable to provide security audit trails. Appellants disagree with the Examiner and submit that at the Examiner's cited passage (Barker, column 17, line 27 – column 18, line 67) Barker only refers to auditing when describing the clean up of filters for a removed client session. For instance, Barker states, "when the Client Session Manager removes a session and/or application from its internal structures, it notifies the Event Distributor via a callback, at which point the Event Distributor removes all filters associated with the session and/or application" (Barker, column 18, lines 10-18). Thus, Barker refers to active auditing of client sessions to facilitate clean up of event filter lists, but does not include anything about providing security audit trails.

Claims 11 and 13-15:

In regard to claim 11, Barker fails to teach that a gateway providing security audit trails comprises the gateway providing access to a logging service. As shown in the arguments above regarding claim 10, Barker fails to teach a gateway providing *security* audit trails. Barker also fails to teach that the gateway providing security audit trails comprises the gateway providing access to a logging service. The Examiner cites passages referring to individual components of Barker's system storing lists of events to storage devices (Barker, column 11, lines 18-60, column 17, line 33-column 18, line 9, and column 41, line 63 – column 42, line 53). However, Appellants submit that individual components using storage devices to maintain their own lists of data does not

equate to or imply a gateway providing access to a logging *service*. Furthermore, no component of Barker's system uses a logging service provided by a gateway when storing event lists to storage devices.

In the Response to Arguments section, the Examiner merely cites the same passages of Barker as cited in the rejection of claim 11 without providing any additional explanation or argument regarding his interpretation regarding the teaches of the cited passages. Appellants maintain that the cited passages do teach a gateway providing access to a logging service. As noted above, individual components storing event lists to their own storage devices fails to teach a gateway providing access to a logging service.

Claim 12:

Regarding claim 12, Barker fails to disclose that the logging service is operable to log an ID of a user that receives each event or sends each request. The Examiner contends that Barker teaches, "the logging service (local data services at the server) is operable to log an ID of a user that sends each request" (parenthesis and underlining in original). The Examiner's interpretation of Barker is incorrect. The Examiner cites the same passage cited in regard to the rejection of claim 11 above, but Appellants note that these passages merely refer to the fact that Barker's system includes the ID of a client application when registering event filters for that client application. Such use of the client application ID does not imply that Barker provides access to a logging service operable to log an ID of a user that receives each event or sends each request. Instead, Barker teaches that his Event Distributor provides an IDL interface for registering filters based in part on an application ID (Barker, column 17, lines 28-30).

In response to the above argument, the Examiner, in the Response to Arguments section of the Final Action, argues that "it is noted that the features upon which Appellant relies, 'the logging service, local data services at the server, is operable to log an ID of a user that sends each request' are not recited in the rejected claim(s)." The Examiner has either misread or misunderstood Appellants' argument. Appellants are not arguing that

the specific phrase is recited in any claim, but instead Appellants are refuting the Examiner's assertion regarding Barker's teachings. Specifically, as argued previously and above, the Examiner contends that Barker teaches a logging service, which the Examiner is presumably equating to Barker's local data services at the server, that is operable to log an ID of a user that sends each request. Appellants' argument is directed to pointing out how the Examiner's interpretation of Barker's teachings is incorrect. Thus, Examiner's Response to Argument regarding the fact that "local data services at the server" is not recited in Appellants' claims is completely irrelevant to Appellants' arguments.

Claim 18:

Regarding claim 18, Barker fails to anticipate wherein requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. Instead, Barker teaches, "SNMP Mediator 160 provides translation between the MIB ASN.1 format and the managed object notation used in this architecture" (Barker, column 11, lines 39-42). The Examiner cites a passage (column 21, line 46 - column 22 line 59) where Barker notes that new managed objects could be added (to his system) that utilize a different protocol and encapsulate that knowledge in the managed object class (Barker, column 22, lines18-20). However, Barker fails to mention the Portable Management Interface (PMI) format. The Examiner is apparently arguing that by simply stating that other formats may be used, Barker is specifically anticipating every other possible format, including PMI. This is clearly an erroneous argument. Appellants assert that Barker fails to disclose that the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects as contended by the Examiner.

In the Response to Arguments section, the Examiner merely repeats the rejection and again cites column 21, line 46 to column 22 line 59 of Barker. Thus, the Examiner fails to provide any additional argument or explanation regarding his contention that Barker specifically anticipates converting requests from the interface definition language

to PMI. Without specific teachings by Barker regarding converting requests into PMI, which is a specific format, Barker clearly fails to *anticipate* such conversions. As the Examiner should be aware, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference" (See, M.P.E.P. 2131). Barker clearly fails to expressly or inherently describe converting requests to PMI.

Claim 19:

Regarding claim 19, Barker fails to disclose wherein requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects. The Examiner cites Barker's SNMP mediator providing translation between the MIB ASN.1 format and the managed object notation used in Barker's architecture. Again, the Examiner's interpretation of Barker is incorrect. Barker teaches the use of SNMP as the communication protocol between element management system and the managed elements (Barker, column 4, lines 43-45). However, as is well known in the art, SNMP is not a platform-specific format, but rather is a network protocol that contains no platform specific features. Thus, Barker fails to teach the requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects as asserted by the Examiner.

Claims 20, 23-26, 28, 35, 36, 39, 42-45, 47, 54 and 55:

Regarding claim 20, Barker does not disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application. In contrast, as shown in the arguments regarding claim 1 above, Barker discloses a method of client-based access control of network elements as a means to restrict access on a command/client basis. The Examiner argues that Barker uses "a naming service that provides individual object level access control so that an agent is granted access to an object on the network to support the IIOP protocol" citing column 8, line 53 – column 9,

line 19 and column 7, lines 47-63. As noted above regarding claim 1, that these passages of Barker only refer to his use of EMAPI, CORBA, Java, C++, and SNMP, but fail to mention anything regarding any sort of access control for any portion of Barker's system. The Examiner has not cited any particular portion in Barker that describes the features the Examiner is attributing to Barker's system. In fact, the Examiner is incorrectly assuming that Barker's use of CORBA and the IIOP protocol includes object level access control such that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

Barker further teaches the use of a single service object "to provide services for a class of managed objects" (underlining added) (Barker, column 14, lines 42-43) and that the EM server "will implement one application-specific service object for each type of physical or logical resource to be managed" (underlining added) (Barker, column 39, lines 60-62). Appellants assert that access control on a command/client basis while using a single service object for *each class* of managed object actually teaches away from determining *on a managed object level* whether or not the manager application is allowed to send a request to the managed object.

Furthermore, as with the rejection of claim 1 described above, Barker fails to disclose that access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects. Instead, as noted above, Barker discloses a method "of *client based* access control of network elements" (emphasis added, Barker, column 30, lines 45-46) that "provides a means to restrict access on a *command/client basis*" (emphasis added, Barker, column 31, lines 10-12). Barker does not describe his access control features as restricting access at the object level. Please refer to Appellants arguments above regarding claim 1 for a more detailed discussion regarding Barker's failure to teach object level access control.

Claim 21 and 40:

Regarding claim 21, Barker fails to teach wherein the gateway is configurable to determine whether each of the managers is authorized to communicate with each of the managed objects. Instead, as noted above regarding claim 2, Barker teaches the use of a single service object "to provide services for a class of managed objects" (Barker, column 14, lines 42-43) and that the EM server "will implement one application-specific service object for each type of physical or logical resource to be managed" (underlining added) (Barker, column 39, lines 60-62). Appellants assert that access control on a command/client basis while using a single service object for each class of managed object actually teaches away from determining on a managed object level whether or not the manager application is allowed to send a request to the managed object. Additionally, Barker discloses client-based access control that provides a means to restrict access on a command/client basis (Barker, column 31, lines 10-12). For a more detailed discussion regarding Barker's failure to teach a gateway configurable to determine whether each manager is authorized to communicate with each managed object, please see Appellants' arguments regarding the rejection of claim 2 above, as they also apply to the rejection of claim 21 as well.

Claim 22 and 41:

Regarding claim 22, Barker fails to disclose a gateway that is configurable to authenticate the managers to receive the events from or to send the request to the managed objects as a function of the identity of the managed object. As noted above regarding claim 3, Barker teaches the use of basic server authentication, SSL, and web server administration including client name and password for access control (Barker, column 8, lines 31-54). Further, Barker discloses client based access control that provides a means to restrict access on a command/client basis (Barker, column 31, lines 10-12). However, basic server authentication and SSL using client names and passwords do not imply authenticating managers as a function of the identity of the managed object. For a more detailed discussion regarding Barker's failure to teach a gateway configurable to authenticate managers to send requests to and receive events from a managed object as

a function of the identity of a managed object, please see Appellants' arguments above regarding the rejection of claim 3, as they apply here as well.

Claim 27 and 46:

In regard to claim 27, Barker fails to disclose wherein the managed objects comprise one or more objects *corresponding* to a telephone network, as asserted by the Examiner. In contrast, as described above, Barker discloses a system client that is connected to a network element and element management system client *through* a public switched telephone network (Barker, column 3, lines 48-53). Additionally, Barker teaches the use of a telephone system network through the computer internet and a telephonic link for a system client to connect to the system server (Barker, column 3, lines 54-62). None of the managed objects in Barker *correspond to a telephone network* themselves, but instead communicate using a telephone network. Thus, Barker does not teach wherein the managed objects comprise one or more objects corresponding to a telephone network. For a more detailed discussion regarding Barker's failure to teach managed objects corresponding to a telephone network, please refer to Appellants' arguments above regarding claim 8, as they apply here as well.

Claim 29 and 48:

Regarding claim 29, the Examiner contends that Barker teaches a gateway that is configurable to provide security audit trails. Appellants disagree with the Examiner and submit that at the Examiner's cited passage (Barker, column 17, line 27 – column 18, line 67) Barker only refers to auditing when describing the clean up of filters for a removed client session. For instance, Barker states, "when the Client Session Manager removes a session and/or application from its internal structures, it notifies the Event Distributor via a callback, at which point the Event Distributor removes all filters associated with the session and/or application" (Barker, column 18, lines 10-18). Thus, Barker refers to active Auditing of client sessions to facilitate clean up of event filter lists, but does not include anything about providing security audit trails.

For a more detailed discussion regarding Barker's failure to teach a gateway configured to provide security audit trails, please refer to Appellants' arguments above regarding claim 10, as they apply here as well.

Claim 30, 32-34, 49 and 51-53:

In regard to claim 30, Barker fails to teach that the gateway providing security audit trails comprises the gateway providing access to a logging service. As shown in the arguments above regarding claims 10 and 29, Barker fails to teach a gateway providing security audit trails. Additionally, Barker also fails to teach that the gateway providing security audit trails comprises the gateway providing access to a logging service. The Examiner cites passages referring to individual components of Barker's system storing lists of events to storage devices (Barker, column 11, lines 18-60, column 17, line 33- column 18, line 9, and column 41, line 63 – column 42, line 53). However, individual components using storage devices to maintain their own lists of data does not equate to a gateway providing access to a logging service.

For a more detailed discussion regarding Barker's failure to teach a gateway providing access to a logging service, please refer to Appellants' arguments above regarding claim 11, as they apply here as well.

Claim 31 and 50:

Regarding claim 31, Barker fails to disclose wherein the logging service is operable to log an ID of a user that receives each event or sends each request. The Examiner contends that Barker teaches, "the logging service (local data services at the server) is operable to log an ID of a user that sends each request" (parenthesis and underlining in original). The Examiner cites the same passage cited in regard to the rejection of claim 30 above, but Appellants note that these passages merely refer to the fact that Barker's system includes the ID of a client application when registering event filters for that client application. However, as noted above, such use of the client application ID does not imply that Barker provides access to a logging service operable to

log an ID of a user that receives each event or sends each request. Instead, Barker teaches that his Event Distributor provides an IDL interface for registering filters based in part on an application ID (Barker, column 17, lines 28-30).

For a more detailed discussion regarding Barker's failure to teach a logging service operable to log an ID of a user that receives each event or sends each request, please refer to Appellants' arguments above regarding claim 12, as they apply here as well.

Claim 37 and 56:

Regarding claim 37, Barker fails to disclose wherein requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects. Instead, as described above, Barker teaches, "SNMP Mediator 160 provides translation between the MIB ASN.1 format and the managed object notation used in this architecture" (Barker, column 11, lines 39-42). The Examiner column 21, line 46 - column 22 line 59 where Barker notes that new managed objects could be added (to his system) that utilize a different protocol and encapsulate that knowledge in the managed object class (Barker, column 22, lines18-20). However, Barker fails to mention the PMI format. The Examiner is apparently arguing that by simply stating that other formats may be used, Barker is specifically anticipating every other possible format, including PMI. This is clearly an incorrect interpretation of Barker's teachings. Therefore, Barker fails to teach that the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects as contended by the Examiner.

For a more detailed discussion regarding Barker's failure to teach that requests are converted from IDL to PMI format prior to delivery, please refer to Appellants' arguments above regarding claim 18, as they apply here as well.

Claim 38 and 57:

Regarding claim 38, Barker fails to disclose wherein requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects. The Examiner cites Barker's SNMP mediator providing translation between the MIB ASN.1 format and the managed object notation used in Barker's architecture. Appellants disagree with the Examiner's interpretation of Barker. As noted above, Barker teaches the use of SNMP as the communication protocol between element management system and the managed elements (Barker, column 4, lines 43-45). As is well known in the art, SNMP is not a platform-specific format, but rather is a network protocol that contains no platform specific features. Thus, Barker fails to teach the requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects as asserted by the Examiner.

Third Ground of Rejection:

Claims 58-63 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Vuong et al. (U.S. Patent 6,430,578) (hereinafter "Vuong"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 58:

Regarding claim 58, contrary to the Examiner's assertion, Vuong fails to disclose a gateway which is coupled to a plurality of managed objects and which is configured to deliver events generated by the managed objects to one or more managers or to deliver requests generated by the managers to one or more of the managed objects. Vuong teaches a naming service that provides unique identifiers and addresses for processes on a computer network. Vuong's name service includes a database of the identifiers and addresses and the name service responds to queries by searching the database and returning any results. (Vuong, Abstract; column 2, lines 7-15).

The Examiner cites column 5, line 57 – column 6, line 23. However, the cited passage describes how Vuong's name service accepts names from agents on the computer network and, after determining whether or not the name is unique, either adds the agent's name to the name service's database or sends a "refuse request" message to the agent. The cited passage does not mention any gateway coupled to a plurality of managed objects. Database entries are not managed objects, as managed objects are understood in the art. Presumably the Examiner interprets Vuong's name service as a gateway. However, Vuong's name service is not coupled to a plurality of managed objects. Instead, Vuong's name service merely handles requests to add names to as well as queries to retrieve information from the name service's database. Even if one could interpret Vuong's name service database as a managed object, which Appellants maintain one cannot, the database is clearly not managed by the requesting agents. Merely requesting that a name and/or address be inserted as an entry into the database does not constitute *managing* the database. Clearly Vuong's name service manages the database. In fact, Vuong very clearly states, "Name Service 112 *maintains* a database holding identification and addressing information" and "the database *controlled by* the Name service is an object-oriented database" (emphasis added, Vuong, column 3, lines 57-63). Thus, Vuong teaches that his name service controls and maintains the database.

Additionally, agents registering their names with Vuong's name service are not managers and do not generate requests to managed objects. Instead, Vuong's agents merely request that their name (and address) be included in the name service's database. Vuong does not teach that an agent registering its name with the name service is a manager generating requests to a managed object.

Vuong also fails to disclose a gateway configurable to provide object-level access control between the managers and the managed objects. The Examiner cites column 2, lines 26-52 and column 6, lines 42-59 of Vuong. The first cited passage provides an introduction to Vuong's name service for "managing names and identities of processes running on a computer network" (Vuong, column 2, lines 26-28). This passage further describes how Vuong's name service includes a receiver that accepts a name from a

process on the computer network and a comparator configured to determine whether the process is a component of the computer management infrastructure for the computer network. The second cited passage (Vuong, column 6, lines 42-59) describes the ability of Vuong's name service to respond to "relatively sophisticated queries." For example, Vuong's query syntax supports prefixes, suffixes, infixes, and full or partial names using wildcards. This passage further describes how registered entities may receive updates or changes made to the name service's database. However, nowhere in either cited passage, nor in fact in the entire Vuong reference, is there any mention of a gateway configured to provide *object-level access control* between managers and managed objects.

Instead, Vuong provides a name service that collects, maintains, and disseminates unique identifiers and addresses for processes on a computer network. Providing identifiers and addresses for processes on a computer network is clearly not the same as providing object-level access control between managers and managed objects. Vuong does not mention any sort of access control in his name service. The Examiner seems to be implying that any form of object-level access necessarily includes object-level access *control* at the individual object level. However, object-level access can be provided with or without imposing object-level access *control*. Vuong does not disclose or complete any form of access control.

Furthermore, Vuong fails to disclose wherein the object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. The Examiner again cites column 2, lines 26-52 and column 6, lines 42-59 of Vuong. However, neither of these passages mentions anything regarding a agent, which the Examiner is presumably interpreting as a manager, being granted access to one database entry, which the Examiner is presumably interpreting as a managed object, while being prevented from interfacing with a different one of the database entries. Instead, the cited passages describe how Vuong's name service responds to queries. Vuong doesn't mention anything regarding preventing access to his database on an entry-level basis.

Claim 59:

Regarding claim 59, Vuong fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application. The Examiner cites column 7, lines 9-32. However, the cited reference has absolutely no relevance to determining, as a function of the identity of a user of the manager application whether or not the manager application is allowed to receive an event generated by or to send a request to one of a plurality of managed object. Instead, the cited reference merely describes how an agent, or other entity on the computer network, can de-register with Vuong's name service and thereby remove its name from the name service's database. The cited reference makes not mention to determining whether or the requesting agent can access a managed object. Even if one interprets the entries of Vuong's database as managed object, which Appellants maintain one cannot, the cited passage still does not disclose anything regarding determining whether or not the de-registering agent can access the database entry. Instead, Vuong teaches only that the name service checks the agent's name against the database and if it is found, the entry is removed.

Vuong also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner contention. The Examiner cites column 8, lines 21-42 of Vuong. Appellants can see no relevance of the cited passage. The cited passage discusses the "various devices and entities" that reside on and communicate over a computer network. Vuong mentions devices and entities such as client computers, data storage devices, modems, printers, hubs, routers, packet switches, hosts, and bridges. The cited passage is, however,

completely silent regarding approving or denying access for a manager application at an individual object level so that the manager application is granted access to one while being prevented from interfacing with a different one of a plurality of managed objects. The Examiner seems to be arguing that merely listing various devices that may reside and communicate on a computer network implies providing such access control at an individual object level. The Examiner is clearly inserting his own assumptions into Vuong's system through hindsight speculation.

Claim 60:

Regarding claim 60, Vuong fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application. The Examiner cites column 7, lines 9-32. However, as noted above regarding claim 59, the cited reference has absolutely no relevance to determining, as a function of the identity of a user of the manager application whether or not the manager application is allowed to receive an event generated by or to send a request to one of a plurality of managed object. Instead, the cited reference merely describes how an agent, or other entity on the computer network, can de-register with Vuong's name service and thereby remove its name from the name service's database. The cited reference makes not mention to determining whether or the requesting agent can access a managed object. Even if one interprets the entries of Vuong's database as managed object, which Appellants maintain one cannot, the cited passage still does not disclose anything regarding determining whether or not the de-registering agent can access the database entry. Instead, Vuong teaches only that the name service checks the agent's name against the database and if it is found, the entry is removed.

Vuong also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted

access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner contention. The Examiner cites column 8, lines 21-42 of Vuong. Appellants can see no relevance of the cited passage. As noted above regarding claim 59, the cited passage discusses the "various devices and entities" that reside on and communicate over a computer network. For a more detailed discussion regarding Vuong's failure to teach access control at an individual object level, please refer to Appellants' arguments above regarding claim 59.

Claim 61:

Regarding claim 61, contrary to the Examiner's assertion, Vuong fails to disclose a gateway coupled to a plurality of managed objects; and which is configured to deliver events generated by the managed objects to one or more managers or to deliver requests generated by the managers to one or more of the managed objects. As described previously, Vuong teaches naming service that provides unique identifiers and addresses for processes on a computer network. Vuong's name service includes a database of the identifiers and addresses and responds to queries by searching the database and returning any results. (Vuong, Abstract; column 2, lines 7-15).

The Examiner cites column 5, line 57 – column 6, line 23. However, the cited passage describes how Vuong's name service accepts names from agents on the computer network and, after determining whether or not the name is unique, either adds the agent's name to the name service's database or sends a "refuse request" message to the agent. The cited passage does not mention any gateway coupled between a plurality of managed objects and a plurality of proxy agent managers. Database entries are not managed objects, as managed objects are understood in the art. Presumably the Examiner interprets Vuong's name service as a gateway. However, Vuong's name service is not coupled between a plurality of managed objects and a plurality of proxy agent managers. Instead, Vuong's name service merely responds to requests to add to and queries to retrieve information from the name service's database. Even if one could interpret

Vuong's name service as a managed object, which Appellants maintain one cannot, the database is clearly not managed by the requesting agents. Furthermore, Vuong's name service does not deliver events generated by any managed objects to the managers. For the Examiner's interpretation to be correct, Vuong's name server should have to deliver events generated by the database entries, which presumably the Examiner equates to managed objects. However, Vuong's database does not generate any events.

Vuong also fails to disclose a gateway configurable to provide object-level access control between the managers and the managed objects. The Examiner cites column 2, lines 26-52 and column 6, lines 42-59 of Vuong. The first cited passage provides an introduction to Vuong's name service for "managing names and identities of processes running on a computer network" (Vuong, column 2, lines 26-28). This passage further describes how Vuong's name service includes a receiver that accepts a name from a process on the computer network and a comparator configured to determine whether the process is a component of the computer management infrastructure for the computer network. The second cited passage (Vuong, column 6, lines 42-59) describes the ability of Vuong's name service to respond to "relatively sophisticated queries." For example, Vuong's query syntax supports prefixes, suffixes, infixes, and full or partial names using wildcards. This passage further describes how registered entities may receive updates or changes made to the name service's database.

Nowhere in either cited passage, nor in fact in the entire Vuong reference, is there any mention of a gateway configured to provide object-level access control between managers and managed objects. Instead, Vuong provides a name service that collects, maintains, and disseminates unique identifiers and addresses for processes on a computer network. Providing identifiers and addresses for processes on a computer network is clearly not the same as providing object-level access control between managers and managed objects. Vuong does not mention any sort of access control in his name service. The Examiner seems to be implying that any form of object-level access necessarily includes object-level access *control*. However, object-level access can be provided with

or without imposing *access control*. In Vuong, no form of access control is disclosed or contemplated.

Furthermore, Vuong fails to disclose wherein the object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. The Examiner again cites column 2, lines 26-52 and column 6, lines 42-59 of Vuong. However, neither of these passages mentions anything regarding a agent, which the Examiner is presumably interpreting as a manager, being granted access to one database entry, which the Examiner is presumably interpreting as a managed object, while being prevented from interfacing with a different one of the database entries. Instead, the cited passages describe how Vuong's name service responds to queries. Vuong doesn't mention anything regarding preventing access to his database on an entry-level basis.

Claim 62:

Regarding claim 62, Vuong fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application. The Examiner cites column 7, lines 9-32. However, as described previously, the cited reference has absolutely no relevance to determining, as a function of the identity of a user of the manager application whether or not the manager application is allowed to receive an event generated by or to send a request to one of a plurality of managed object. Instead, the cited reference merely describes how an agent, or other entity on the computer network, can de-register with Vuong's name service and thereby remove its name from the name service's database. The cited reference makes not mention to determining whether or the requesting agent can access a managed object. For a more detailed discussion of Vuong's failure to teach access control as a function of the identity

of the user of the manager application, please see Appellants' arguments above regarding claim 59.

Vuong also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner contention. The Examiner cites column 8, lines 21-42 of Vuong. Appellants can see no relevance of the cited passage. The cited passage discusses the "various devices and entities" that reside on and communicate over a computer network. Vuong mentions devices and entities such as client computers, data storage devices, modems, printers, hubs, routers, packet switches, hosts, and bridges. The cited passage is, however, completely silent regarding approving or denying access for a manager application at an individual object level so that the manager application is granted access to one while being prevented from interfacing with a different one of a plurality of managed objects. For a more detailed discussion regarding Vuong's failure to teach access control at an individual object level, please refer to Appellants' arguments above regarding claim 59.

Claim 63:

Regarding claim 63, Vuong fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application. The Examiner cites column 7, lines 9-32. However, as noted above regarding claim 62, the cited reference has absolutely no relevance to determining, as a function of the identity of a user of the manager application whether or not the manager application is allowed to receive an event generated by or to send a request to one of a plurality of managed object. Instead, the cited reference merely describes how an agent, or other entity on the computer network, can de-register with Vuong's name service and thereby remove its

name from the name service's database. The cited reference makes not mention to determining whether or the requesting agent can access a managed object.

Vuong also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner contention. The Examiner cites column 8, lines 21-42 of Vuong. Appellants can see no relevance of the cited passage. The cited passage discusses the "various devices and entities" that reside on and communicate over a computer network. The cited passage is, however, completely silent regarding approving or denying access for a manager application at an individual object level so that the manager application is granted access to one while being prevented from interfacing with a different one of a plurality of managed objects.

For a more detailed discussion regarding Vuong's failure to teach access control at an individual object level and Vuong's failure to teach access control as a function of the identity of the user of the manager application, please refer to Appellants' arguments above regarding claims 62 and 59.

Fourth Ground of Rejection:

Claims 58-63 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Spencer (U.S. Patent 6,253,243) (hereinafter "Spencer"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 58:

Regarding claim 58, Spencer fails to disclose a gateway configurable to provide object-level access control between the managers and the managed objects to receive the

events from or to send the requests to the managed objects, contrary to the Examiner's assertion. The Examiner cites a passage (column 5, lines 46-65) where Spencer describes how a user-developed management application 300 communicates with MIS server 306 via a portable management interface (PMI) 302. Spencer describes how PMI 302 is an object-oriented interface that provides access to management information. The cited passage does not teach anything about a gateway providing object-level access control between managers and managed objects. The Examiner has not provided any argument or explanation regarding his interpretation of the cited passage.

Spencer further fails to disclose wherein the object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects, contrary to the Examiner's contention. The Examiner cites column 7, lines 35-57 of Spencer. However, the cited passage teaches how Spencer's SNMP trap system extracts the IP address from an <agent_addr> field of the SNMP trap Protocol Data Unit (PDU). The PDU is the format for SNMP trap data in Spencer's system. After extracting the IP address, Spencer's system determines if there is an object configured to represent that agent system. If such an object is found, the trap's originating system's cmipsnmpProxyAgent instance is set as the source object instance for the trap alarm. Thus, the cited passage not only fails to mention anything about object-level access control, it has no relevance to access control. Spencer does not teach anything about providing object-level access control at the individual object level so that a manager is granted access to one managed object while being prevented from interfacing with a different one of the managed objects.

Claim 59:

Regarding claim 59, contrary to the Examiner's assertion, Spencer fails to disclose sending an identity of a user of a manager application to a gateway. The Examiner cites column 7, lines 35-67 of Spencer. However the cited passage makes no mention of sending an identity of a *user of a manager* application to a gateway. Instead,

the cited passage describes how Spencer's system uses an IP address to locate a proxy agent object to represent a SNMP trap's agent system. Nowhere does Spencer mention sending an identity of a user of a manager application to a gateway.

Additionally, Spencer fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, contrary to the Examiner's contention. The Examiner cites column 5, line 53 to column 6, line 13. However, the cited passage does not teach or even mention determining on a managed object level whether or note the manager application is allowed to receive an event generated by or to send a request to one of the plurality of managed objects as a function of the identity of the user of the manager application. Instead, the cited passage describes how a managed application 300 communicates with an MIS server according to the portable management interface and how the portable management interface is able to access managed object instance state information, class schema, and event services. Spencer does not discuss or mention anything about an identity for a user of a manager application. Nor does Spencer mention determining whether or not the manager application can send requests or send events to managed objects as a function of the identity of the user of the manager application.

Spencer also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner assertion. The Examiner again cites column 7, lines 35-67 of Spencer. However, as noted above, this passage does not mention any sort of object-level access control. Nowhere does Spencer mention anything regarding approving or denying the manager application access to receive an event or send a request at the individual object level. The

cited passage fails to mention any sort of access control whatsoever. The Examiner has clearly misunderstood or misinterpreted the teachings of Spencer.

Claim 60:

Regarding claim 60, contrary to the Examiner's assertion, Spencer fails to disclose sending an identity of a user of a manager application to a gateway. The Examiner cites column 7, lines 35-67 of Spencer. However the cited passage makes no mention of sending an identity of a *user of a manager* application to a gateway. Instead, the cited passage describes how Spencer's system uses an IP address to locate a proxy agent object to represent a SNMP trap's agent system. Nowhere does Spencer mention sending an identity of a user of a manager application to a gateway.

Additionally, Spencer fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, contrary to the Examiner's contention. The Examiner cites column 5, line 53-column 6, line 13. However, as noted above regarding claim 59, the cited passage does not teach or even mention determining on a managed object level whether or note the manager application is allowed to receive an event generated by or to send a request to one of the plurality of managed objects as a function of the identity of the user of the manager application. For a more detailed discussion regarding Spencer's failure to teach managed object level access control as a function of the identity of a user of a manager application, please see Appellants' arguments above regarding claim 59.

Spencer also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner

assertion. The Examiner again cites column 7, lines 35-67 of Spencer. However, as noted above, this passage does not mention any sort of object-level access control. For a more detailed discussion, please see Appellants' arguments above regarding claim 59.

Claim 61:

Regarding claim 61, Spencer fails to disclose a gateway configurable to provide object-level access control between the managers and the managed objects to receive the events from or to send the requests to the managed objects, contrary to the Examiner's assertion. The Examiner cites a passage (column 5, lines 46-65) where Spencer describes how a user-developed management application 300 communicates with MIS server 306 via a portable management interface (PMI) 302. Spencer describes how PMI 302 is an object-oriented interface that provides access to management information. The cited passage does not teach anything about a gateway providing object-level access control between managers and managed objects. The Examiner has not provided any argument or explanation regarding his interpretation of the cited passage.

Spencer further fails to disclose wherein the object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects, contrary to the Examiner's contention. The Examiner cites column 7, lines 35-57 of Spencer. However, the cited passage teaches how Spencer's SNMP trap system extracts the IP address from an <agent_addr> field of the SNMP trap Protocol Data Unit (PDU). The PDU is the format for SNMP trap data in Spencer's system. After extracting the IP address, Spencer's system determines if there is an object configured to represent that agent system. If such an object is found, the trap's originating system's cmipsnmpProxyAgent instance is set as the source object instance for the trap alarm. Thus, the cited passage not only fails to mention anything about object-level access control, it has no relevance to object-level access control. The cited passage does not teach anything about providing object-level access control at the

individual object level so that a manager is granted access to one managed object while being prevented from interfacing with a different one of the managed objects.

Furthermore, Spencer fails to disclose wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which a manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control. The Examiner cites column 5, lines 2-34 and lines 47-67. However, neither of the cited passages makes any reference whatsoever regarding a gateway using a single SAP object. Nor does the cited reference mention anything about inserting a user name in a request message to enforce object-level access control. In fact, nowhere does Spencer mention including user identification in a request message or regarding a single SAP object.

Claim 62:

Regarding claim 62, contrary to the Examiner's assertion, Spencer fails to disclose sending an identity of a user of a manager application to a gateway. The Examiner cites column 7, lines 35-67 of Spencer. However the cited passage makes no mention of sending an identity of a *user of a manager* application to a gateway. Instead, the cited passage describes how Spencer's system uses an IP address to locate a proxy agent object to represent a SNMP trap's agent system. Nowhere does Spencer mention sending an identity of a user of a manager application to a gateway.

Additionally, Spencer fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, contrary to the Examiner's contention. The Examiner cites column 5, line 53 to column 6, line 13. However, the cited passage does not teach or even mention determining on a managed object level whether or note the manager application is allowed to receive an event generated by or to send a request to one of the plurality of managed objects as a function

of the identity of the user of the manager application. For a more detailed discussion of this argument, please refer to Appellants arguments above regarding claim 61.

Spencer also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner assertion. The Examiner again cites column 7, lines 35-67 of Spencer. However, as noted above, this passage does not mention any sort of object-level access control. The cited passage further fails to mention anything regarding approving or denying the manager application access to receive an event or send a request at the individual object level. The cited passage fails to mention any sort of access control whatsoever.

Furthermore, as discussed above regarding claim 61, Spencer fails to disclose wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which a manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control. The Examiner cites column 5, lines 2-34 and lines 47-67. However, neither of the cited passages makes any reference whatsoever regarding a gateway using a single SAP object. Nor does the cited reference mention anything about inserting a user name in a request message to enforce object-level access control. In fact, nowhere does Spencer mention including user identification in a request message or regarding a single SAP object.

Claim 63:

Regarding claim 63, contrary to the Examiner's assertion, Spencer fails to disclose sending an identity of a user of a manager application to a gateway. The Examiner cites column 7, lines 35-67 of Spencer. However the cited passage makes no mention of sending an identity of a *user of a manager* application to a gateway. Instead, the cited passage describes how Spencer's system uses an IP address to locate a proxy

agent object to represent a SNMP trap's agent system. Nowhere does Spencer mention sending an identity of a user of a manager application to a gateway.

Additionally, Spencer fails to disclose determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, contrary to the Examiner's contention. The Examiner cites column 5, line 53 to column 6, line 13. However, the cited passage does not teach or even mention determining on a managed object level whether or note the manager application is allowed to receive an event generated by or to send a request to one of the plurality of managed objects as a function of the identity of the user of the manager application.

Spencer also fails to disclose whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects, contrary to the Examiner assertion. The Examiner again cites column 7, lines 35-67 of Spencer. However, as noted above, this passage does not mention any sort of object-level access control.

Furthermore, as discussed above regarding claim 62, Spencer fails to disclose wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which a manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control. The Examiner cites column 5, lines 2-34 and lines 47-67. However, neither of the cited passages makes any reference whatsoever regarding a gateway using a single SAP object. Nor does the cited reference mention anything about inserting a user name in a request message to enforce object-level access control.

For a more detailed discussion regarding Spencer's failure to teach the limitations of claim 63, please refer to the arguments above regarding claims 61 and 62.

Fifth Ground of Rejection:

Claims 58-63 stand finally rejected under 35 U.S.C. § 102(e) as being anticipated by Barker. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claim 58:

Regarding claim 58, Barker does not anticipate a gateway that is configurable to provide object-level access control between the managers and the managed objects, wherein said object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects. Instead, as noted above, Barker discloses a system for "access control based on client name and password" (Barker, column 8, lines 45-46). Barker describes this as "a method of *client based* access control of network elements" (Emphasis added, Barker, column 30, lines 45-46). Further, Barker summarizes his access control features with "the *client based access control* ... provides a means to restrict access on a *command/client basis*", not at the object level. (emphasis added, Barker, column 31, lines 10-12). Please refer to the remarks above regarding claim 1 for a more detailed discussion regarding this argument. Furthermore, since claim 58 recites similar limitations as those recited in claim 1, the arguments presented above regarding the rejection of claim 1 in view of Barker also apply to claim 58 with equal force. Thus, Barker does not teach object-level access control between the managers and the managed objects.

Claim 59:

Regarding claim 59, the Examiner states, "Barker teaches... wherein the gateway in configured to ... determine on a managed object level whether or not the manager

application is allowed to send a request to the managed object as a function of the user of the manager application." Appellants disagree with the Examiner's interpretation of Barker and submit that Barker fails to anticipate determining on a managed object level whether or not the manager application is allowed to send a request to the managed object. In contrast, as shown in the arguments regarding claim 1 above, Barker discloses a method of client based access control of network elements as a means to restrict access on a command/client basis. For a more detailed discussion regarding this argument, please refer to the remarks above regarding the rejection of claim 20. Furthermore, as claim 59 recites limitations similar to those recited in claim 20, the arguments presented above regarding claim 20 apply to claim 59 with equal force.

Claim 60:

Regarding claim 60, the Examiner states, "Barker teaches... wherein the gateway in configured to ... determine on a managed object level whether or not the manager application is allowed to send a request to the managed object as a function of the user of the manager application." Appellants disagree with the Examiner's interpretation of Barker and submit that Barker fails to anticipate determining on a managed object level whether or not the manager application is allowed to send a request to the managed object. In contrast, as shown in the arguments regarding claim 1 above, Barker discloses a method of client based access control of network elements as a means to restrict access on a command/client basis. For a more detailed discussion regarding this argument, please refer to the remarks above regarding the rejection of claim 59. Furthermore, the arguments presented above regarding claims 20 and 59 apply to claim here with equal force.

Claim 61:

Regarding claim 61, Barker does not anticipate a gateway that is configurable to provide object-level access control between the managers and the managed objects, wherein said object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being

prevented from interfacing with a different one of the managed objects. Instead, as noted above regarding claims 1 and 58, Barker discloses a system for "access control based on client name and password" (Barker, column 8, lines 45-46). Barker describes this as "a method of *client based* access control of network elements" (Emphasis added, Barker, column 30, lines 45-46). Further, Barker summarizes his access control features with "the *client based access control* ... provides a means to restrict access on a *command/client basis*", not at the object level (emphasis added, Barker, column 31, lines 10-12). Please refer to the remarks above regarding claims 1 and 58 for a more detailed discussion regarding this argument.

Additionally, Barker fails to disclose wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which a manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control. Nowhere does Barker mention inserting a user name in a request message to enforce object-level access control. The Examiner cites column 8, line 53 to column 9, line 33, specifically referring to Barker's naming service. However, as noted above, this passage of Barker only refers to his use of EMAPI, CORBA, Java, C++, and SNMP, but fails to mention anything regarding any sort of access control for any portion of Barker's system and further fails to mention anything regarding a singleton SAP object that allows the insertion of a user name in a request message. The Examiner has not cited any particular portion in Barker that describes the features the Examiner is attributing to Barker's system. In fact, the Examiner is incorrectly assuming that Barker's use of CORBA and the IIOP protocol includes object level access control such that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

Claim 62:

Regarding claim 62, the Examiner states, "Barker teaches... wherein the gateway in configured to ... determine on a managed object level whether or not the manager application is allowed to send a request to the managed object as a function of the user of

the manager application." Appellants disagree with the Examiner's interpretation of Barker and submit that Barker fails to anticipate determining on a managed object level whether or not the manager application is allowed to send a request to the managed object. In contrast, as shown above, Barker discloses a method of client based access control of network elements as a means to restrict access on a command/client basis. For a more detailed discussion regarding this argument, please refer to the remarks above regarding the rejection of claim 20. Furthermore, as claim 62 recites limitations similar to those recited in claim 20, the arguments presented above regarding claim 20 apply to claim 62 with equal force.

Additionally, Barker fails to disclose wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which the manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control. For a more detailed discussion regarding Barker's failure to teach a gateway using a singleton SAP object, please see Appellants' arguments above regarding claim 61.

Claim 63:

Regarding claim 63, the Examiner states, "Barker teaches... wherein the gateway in configured to ... determine on a managed object level whether or not the manager application is allowed to send a request to the managed object as a function of the user of the manager application." Appellants disagree with the Examiner's interpretation of Barker and submit that Barker fails to anticipate determining on a managed object level whether or not the manager application is allowed to send a request to the managed object. For a more detailed discussion regarding this argument, please see the discussion of claims 61 and 62 above.

Additionally, Barker fails to disclose wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which the manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-

level access control. For a more detailed discussion regarding Barker's failure to teach a gateway using a singleton SAP object, please see Appellants' arguments above regarding claims 61 and 62.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-63 was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge the appeal brief fee of $500.00 and any other fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-48400/RCK. This Appeal Brief is submitted with a return receipt postcard.

Respectfully submitted,

Robert C. Kowert
Reg. No. 39,255
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Date: July 12, 2005

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. A network management system, comprising:

a gateway which is coupled to a plurality of managed objects and which is configured to deliver events generated by the managed objects to one or more managers or to deliver requests generated by the managers to one or more of the managed objects; and

a platform-independent interface to the gateway, wherein the gateway is configurable to communicate with the managers through the platform-independent interface to deliver the events or requests;

wherein the gateway is configurable to provide object-level access control between the managers and the managed objects to receive the events from or to send the requests to the managed objects, wherein said object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects.

2. The network management system of claim 1, wherein the gateway is configurable to determine whether each of the managers is authorized to communicate with each of the managed objects.

3. The network management system of claim 1, wherein the gateway is configurable to authenticate the managers to receive the events from or to send the requests to the managed objects as a function of the identity of the managed object.

4. The network management system of claim 1, wherein the gateway is configurable to authenticate the managers to receive the events or send the requests as a

function of user IDs entered by users of the managers.

5. The network management system of claim 1, wherein the events or requests are delivered by the gateway through the platform-independent interface according to Internet Inter-Object Protocol (IIOP).

6. The network management system of claim 1, wherein the platform-independent interface to the gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to the managed objects across a plurality of platforms and across a plurality of programming languages.

7. The network management system of claim 6, wherein the interface definition language comprises OMG IDL.

8. The network management system of claim 1, wherein the managed objects comprise one or more objects corresponding to a telephone network.

9. The network management system of claim 1, wherein the managed objects comprise an object corresponding to a telecommunications device.

10. The network management system of claim 1, wherein the gateway is configurable to provide security audit trails.

11. The network management system of claim 10, wherein the gateway providing security audit trails comprises the gateway providing access to a logging service.

12. The network management system of claim 11, wherein the logging service is operable to log an ID of a user that receives each event or sends each request.

13. The network management system of claim 11, wherein the logging service is operable to log an ID of the managed object that is the source of each event or the target of each request.

14. The network management system of claim 11, wherein the logging service is operable to log a time at which each event or request is generated.

15. The network management system of claim 11, wherein the logging service is operable to log a time at which each event or request is delivered.

16. The network management system of claim 1, wherein the requests comprise a query for information concerning one of the managed objects.

17. The network management system of claim 1, wherein the requests comprise a command to set one or more parameters of one of the managed objects.

18. The network management system of claim 1, wherein the requests are converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed objects.

19. The network management system of claim 1, wherein the requests are converted from the interface definition language to a platform-specific format prior to delivery to the managed objects.

20. A network management method, comprising:

sending an identity of a user of a manager application to a gateway, wherein the gateway is configurable to communicate with the manager application through a platform-independent interface;

determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects; and

delivering the event to the manager application or the request to the managed object if the manager access is approved.

21. The network management method of claim 20, wherein the gateway is configurable to determine whether the manager is authorized to communicate with the managed object.

22. The network management method of claim 20, wherein the gateway is configurable to authenticate the manager to receive the event or send the request as a function as the identity of the managed object generating the event or receiving the request.

23. The network management method of claim 20, wherein the gateway is configurable to authenticate the manager to receive the event or send the request as a function of a user ID entered by the user of the manager.

24. The network management method of claim 20, wherein the event or request is delivered by the gateway through the platform-independent interface according to Internet Inter-Object Protocol (IIOP).

25. The network management method of claim 20, wherein the platform-independent interface to the gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to the managed objects across a plurality of platforms and across a plurality of programming languages.

26. The network management method of claim 25, wherein the interface definition language comprises OMG IDL.

27. The network management method of claim 20, wherein the managed object comprises an object corresponding to a telephone network.

28. The network management method of claim 20, wherein the managed object comprises an object corresponding to a telecommunications device.

29. The network management method of claim 20, wherein the gateway is configurable to provide security audit trails.

30. The network management method of claim 29, wherein the gateway providing security audit trails comprises the gateway providing access to a logging service.

31. The network management method of claim 30, wherein the logging service is operable to log an ID of a user that receives the event or sends the request.

32. The network management method of claim 30, wherein the logging service is operable to log an ID of the managed object that is the source of the event or the target of the request.

33. The network management method of claim 30, wherein the logging service is operable to log a time at which the event or request is generated.

34. The network management method of claim 30, wherein the logging service is operable to log a time at which the event or request is delivered.

35. The network management method of claim 20, wherein the request comprises a query for information concerning the managed object.

36. The network management method of claim 20, wherein the request comprises a command to set one or more parameters of the managed object.

37. The network management method of claim 20, wherein the request is converted from the interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed object.

38. The network management method of claim 20, wherein the request is converted from the interface definition language to a platform-specific format prior to delivery to the managed object.

39. A carrier medium comprising program instructions for network management, wherein the program instructions are computer-executable to perform:

sending an identity of a user of a manager application to a gateway, wherein the gateway is configurable to communicate with the manager application through a platform-independent interface;

determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed

objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects; and

delivering the event to the manager application or the request to the managed object if the manager access is approved.

40. The carrier medium of claim 39, wherein the gateway is configurable to determine whether the manager is authorized to communicate with the managed object.

41. The carrier medium of claim 39, wherein the gateway is configurable to authenticate the manager to receive the event or send the request as a function as the identity of the managed object generating the event or receiving the request.

42. The carrier medium of claim 39, wherein the gateway is configurable to authenticate the manager to receive the event or send the request as a function of a user ID entered by the user of the manager.

43. The carrier medium of claim 39, wherein the event or request is delivered by the gateway through the platform-independent interface according to Internet Inter-Object Protocol (IIOP).

44. The carrier medium of claim 39, wherein the platform-independent interface to the gateway is expressed in an interface definition language, and wherein the interface definition language comprises a language for defining interfaces to the managed objects across a plurality of platforms and across a plurality of programming languages.

45. The carrier medium of claim 44, wherein the interface definition language comprises OMG IDL.

46. The carrier medium of claim 39, wherein the managed object comprises an object corresponding to a telephone network.

47. The carrier medium of claim 39, wherein the managed object comprises an object corresponding to a telecommunications device.

48. The carrier medium of claim 39, wherein the gateway is configurable to provide security audit trails.

49. The carrier medium of claim 48, wherein the gateway providing security audit trails comprises the gateway providing access to a logging service.

50. The carrier medium of claim 49, wherein the logging service is operable to log an ID of a user that receives the event or sends the request.

51. The carrier medium of claim 49, wherein the logging service is operable to log an ID of the managed object that is the source of the event or the target of the request.

52. The carrier medium of claim 49, wherein the logging service is operable to log a time at which the event or request is generated.

53. The carrier medium of claim 49, wherein the logging service is operable to log a time at which the event or request is delivered.

54. The carrier medium of claim 39, wherein the request comprises a query for information concerning the managed object.

55. The carrier medium of claim 39, wherein the request comprises a command to set one or more parameters of the managed object.

56. The carrier medium of claim 39, wherein the request is converted from the

interface definition language to a Portable Management Interface (PMI) format prior to delivery to the managed object.

57. The carrier medium of claim 39, wherein the request is converted from the interface definition language to a platform-specific format prior to delivery to the managed object.

58. A network management system, comprising:

a gateway which is coupled to a plurality of managed objects and which is configured to deliver events generated by the managed objects to one or more managers or to deliver requests generated by the managers to one or more of the managed objects; and

a platform-independent interface to the gateway, wherein the gateway is configurable to communicate with the managers through the platform-independent interface to deliver the events or requests;

wherein the gateway is configurable to provide object-level access control between the managers and the managed objects to receive the events from or to send the requests to the managed objects, wherein said object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects, and wherein the managers use a request Service Access Point (SAP) for requests and responses.

59. A network management method, comprising:

sending an identity of a user of a manager application to a gateway, wherein the gateway is configurable to communicate with the manager application through a platform-independent interface;

determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application, whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects; and

delivering the event to the manager application or the request to the managed object if the manager access is approved;

wherein the manager application uses a request Service Access Point (SAP) for requests and responses.

60. A carrier medium, comprising program instructions for network management, wherein the program instructions are computer-executable to perform:

sending an identity of a user of a manager application to a gateway, wherein the gateway is configurable to communicate with the manager application through a platform-independent interface;

determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application,

whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects; and

delivering the event to the manager application or the request to the managed object if the manager access is approved;

wherein the manager application uses a request Service Access Point (SAP) for requests and responses.

61. A network management system, comprising:

a gateway which is coupled to a plurality of managed objects and which is configured to deliver events generated by the managed objects to one or more managers or to deliver requests generated by the managers to one or more of the managed objects;

a platform-independent interface to the gateway, wherein the gateway is configurable to communicate with the managers through the platform-independent interface to deliver the events or requests;

wherein the gateway is configurable to provide object-level access control between the managers and the managed objects to receive the events from or to send the requests to the managed objects, wherein said object-level access control is provided at the individual object level so that one of the managers is granted access to one of the managed objects while being prevented from interfacing with a different one of the managed objects; and

wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which a manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control.

62. A network management method, comprising:

sending an identity of a user of a manager application to a gateway, wherein the gateway is configurable to communicate with the manager application through a platform-independent interface;

determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application,

whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects; and

delivering the event to the manager application or the request to the managed object if the manager access is approved; and

wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which the manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control.

63. A carrier medium, comprising program instructions for network management, wherein the program instructions are computer-executable to perform:

sending an identity of a user of a manager application to a gateway, wherein the gateway is configurable to communicate with the manager application through a platform-independent interface;

determining on a managed object level whether or not the manager application is allowed to receive an event generated by one of a plurality of managed objects or to send a request to the one of the plurality of managed objects as a function of the identity of the user of the manager application,

whereby access for the manager application to receive the event or send the request is approved or denied for said one of the plurality of managed objects at the individual object level so that the manager application is granted access to one of the plurality of managed objects while being prevented from interfacing with a different one of the plurality of managed objects; and

delivering the event to the manager application or the request to the managed object if the manager access is approved; and

wherein the gateway uses a singleton SAP object that shares all ProxyAgents through which the manager deals with a managed object and allows the insertion of the user name in the request message to enforce object-level access control.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.